Software Tech News

The DoD Source for Software Technology Information.

Vol. 3- No. 1

Special STC '99 issue!

In This Issue:

Read additional Collaborative Technique articles at:

www.dacs.dtic.mil/awareness/newsletters/listing.shtml

DoD Data & Analysis Center for Software
http://www.dacs.dtic.mil

Win Win: A System for Negotiating Requirements

by Ellis Horowitz, Joo H. Lee, and June Sup Lee

Introduction

Win Win is a computer program that aids in the capture, negotiation, and coordination of requirements for a large system. It assumes that a group of people, called stakeholders, have signed on with the express purpose of discussing and refining the requirements of their proposed system. The system can be of any type. Win Win contains facilities for:

- capturing the desires (win conditions) of the stakeholders
- organizing the terminology so that stakeholders are using the same terms in the same way
- expressing disagreements or issues needing resolution
- offering options as potential solutions
- negotiating agreements which resolve the issues
- using third party tools to enlighten or resolve issues
- producing a requirements document that summarizes the current state of the proposed system
- creating documents that support multimedia and hyperlinks
- tracing the ways by which requirements decisions were reached
- checking the completeness and consistency of requirements

Win Win Functionality

A Simple Scenario

WinWin offers a group of users a great deal of functionality, but as a result some planning is useful before getting started. In this section we offer a simple scenario for how users of WinWin might begin their work.

Continued on page 14

Collaborative Techniques

The WebMe Data Visualization Tool

Roseanne Tesoriero and *Marvin V. Zelkowitz*
University of Maryland *Fraunhofer Center for Experimental Software Engineering*

Introduction

The ubiquity of the World Wide Web and the increased presence of the Internet in today's marketplace have led to an environment where software is being developed in geographically distributed locations. This type of distributed development environment raises several new challenges in the area of software project management.

Measurement data has been used to control and improve the software process and its products. For example, the Software Engineering Laboratory of NASA's Goddard Space Flight Center (NASA/SEL) has been using the Quality Improvement Paradigm along with the Goal/Question/Metric paradigm to help evaluate, control and improve software processes and products for over twenty years [1]. The use of measurement data to build and package experience is a key aspect in this approach. One way in which experience is packaged is in a baseline model. A baseline model is created by clustering project data from similar projects together to describe the expected behavior for this class of projects. Management can then compare a new project with this baseline model in order to better understand deviations from the baseline. For example, a project with too few errors per line of code may either represent a superior development process or insufficient testing.

When dealing with a distributed development environment, the use of measurement data to build knowledge and experience for software project management presents several new challenges. When development occurs at physically separated locations, the data may be geographically separate as well. If several organizations are involved in the development (such as with subcontract management), the organizational cultures may be different and the data may be collected using different collection mechanisms leading to variations in the formats. Creating baselines and comparing different projects becomes more difficult due to the differing interpretations to the collected data. When several organizations are involved in the development, privacy becomes an issue. Each organization does not want to give the others unlimited access to its proprietary data.

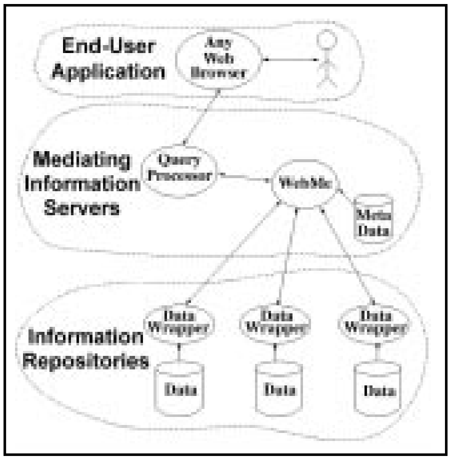

Figure 1. WebMe Architecture

The Web Measurement Environment (WebME) has been developed to address the challenges of using measurement data for experience-based project management in a distributed development environment. This short discussion explains how the techniques incorporated into the WebME system allow project managers to use data from a distributed environment. A brief overview of the WebME system and a description of the analysis technique used to build a baseline from similar projects are given.

WebME solution

The WebME system was designed to build baseline models from data collected in geographically distributed environments. In particular, it can be used to combine and analyze time-series data collected from software projects.

The architecture of the WebME system is shown in Figure 1.

Continued on page 3

Continued from page 2

The WebME system uses a mediated architecture, which horizontally partitions the structure of the architecture into three layers: the information repositories, mediating information servers and the end-user application layer. In the WebME context, the end-user submits a request through a Web browser. The request is directed to the appropriate WebME mediator. The WebME mediator uses its metadata to determine which information repositories to access, obtains the data from the information repositories, and combines the data into an answer for the user query. The results are sent back to the Web browser for presentation.

Metadata is created by using the WebME scripting language. The WebME scripting language allows the system configurer to specify the structure of the architecture, the data that will be viewable with the system, and the access methods for the data. The configurer of the system creates a script containing the definitions for the system. The script is processed, checked for consistency and the results are stored and made available to the mediator. Because the owners of the data are responsible for providing the access methods, access to the data can be limited. Because the metadata contains information about the physical location and context information about the data at remote repositories, data collected at geographically separated locations and stored in different formats can be combined consistently.

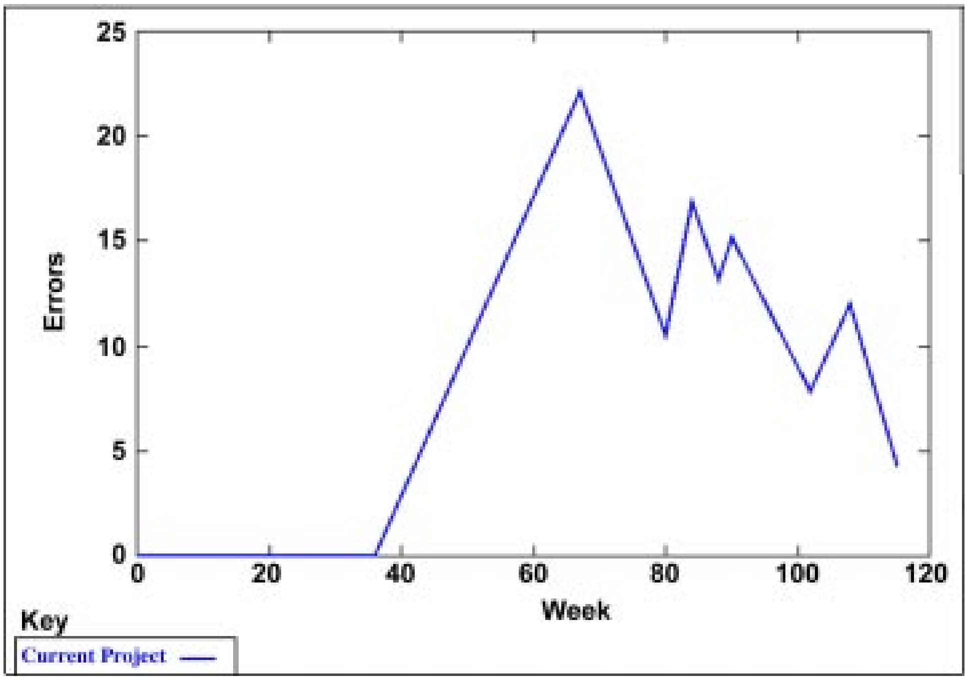

Figure 2. Characteristic Curve of Project under Study

It is important not only to combine the data in a consistent manner, but to use the data to support project management. The analysis technique supported in WebME [2] uses data from past similar projects to build a baseline of expected behavior for an attribute over time. First, a characteristic curve is generated for the project of interest (Figure 2). The characteristic curve describes the behavior of an attribute over the project lifecycle. The baseline model for the attribute is created using the characteristic curves of projects similar to the project of interest (Figure 3).

The baseline model can be used to support project management in several ways. For example, if a project is performing differently from the baseline, it serves a signal for the project manager. In Figure 4, the baseline model of the number of errors per week and the current project's errors per week are plotted. The current project has encountered more errors than expected. The project manager should investigate to determine the cause of the deviation. If the deviation is undesirable, corrective actions can be taken.

Continued on page 4

The WebMe Data Visualization Tool

Continued from page 3

The baseline model may also be used to determine the impact of process changes. If a project manager has implemented a process change, comparing the project performance against the baseline can assist in the understanding of how the process change has affected the attribute behavior of a project (Figure 5). Assume the project manager has incorporated code inspections into the development process to uncover errors earlier in the development cycle. Comparing the current project's characteristic curve for errors with the previous baseline model demonstrates the impact of the process change. In Figure 5, the errors for the current project are discovered earlier than in the baseline model.

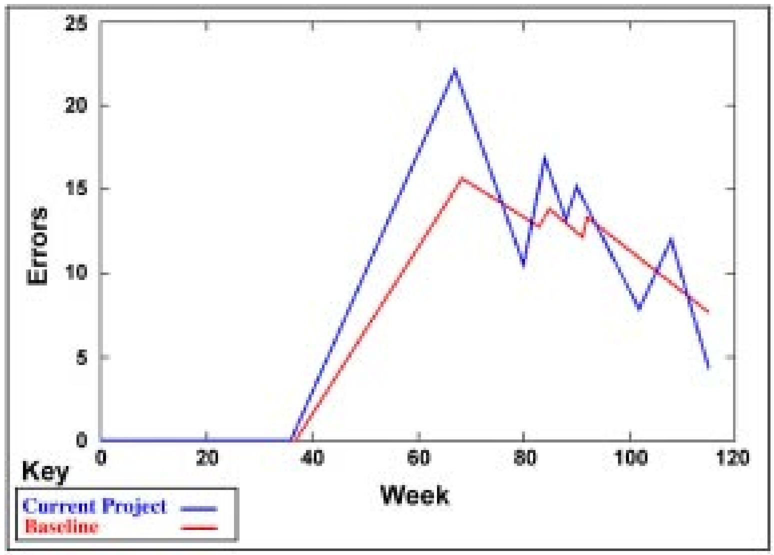

Figure 3. Baseline Characteristic Curve of Related Projects

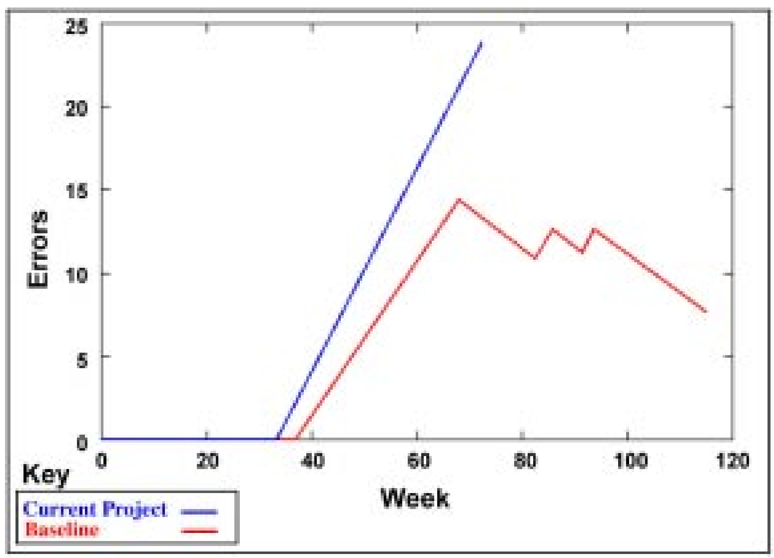

Figure 4. Project that is "Out of Scope"

Author Contact Information

Prof. Roseanne Tesoriero
Department of Computer Science, 4121 A V Williams
University of Maryland, College Park MD 20742

Dr. Marvin Zelkowitz
Fraunhofer Center for Experimental Software Engineering
University of Maryland, College Park MD 20742

mvz@cs.umd.edu

Status

The WebME system has been implemented and a prototype version is available at ftp://ftp.cs.umd.edu/pub/sel/roseanne/webme

Continued on page 5

Collaborative Software Engineering on the Web: Introducing WebDAV

E. James Whitehead, Jr. - University of California, Irvine

Introduction

Software development today often takes place among multiple groups of software engineers who are geographically dispersed. A key challenge in supporting these distributed software development teams is making the software remotely accessible and editable via the Internet, addressing issues such as overwrite prevention, security, authentication, access control, and reliable operation across high-latency network connections.

The Web Distributed Authoring and Versioning (WebDAV) working group of the Internet Engineering Task Force (IETF) has taken on the challenge of supporting collaborative software engineering on the Web by extending the core network protocol of the Web, the Hypertext Transfer Protocol (HTTP) [4], to support remote software development. A team at the University of California, Irvine, working under a grant from the Defense Advanced Research Projects Agency (DARPA) Evolutionary Development of Complex Systems program, brought together interested parties from academia and industry to form the IETF working group and develop the WebDAV Distributed Authoring Protocol.

WebDAV provides many benefits:

- Development teams can collaboratively develop software artifacts in-place on the Web, using locking to prevent overwrite conflicts. Due to the distributed nature of the Web, these work groups can have members from within the same organization, or across organizational boundaries.

Continued on page 6

Continued from page 4

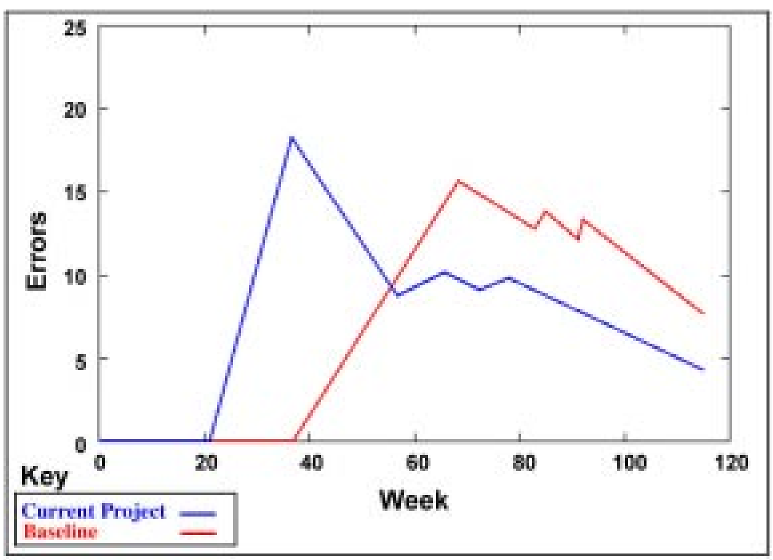

Figure 5. Corrective Action Taken on Project that is “Out of Scope”

References

[1] Basili V., M. Zelkowitz, F. McGarry, J. Page, S. Waligora, and R. Pajerski, SEL’s Software Process-Improvement Program, *IEEE Software* 12, 6 (1995) 83-87.

[2] Tesoriero R. and M. V. Zelkowitz, “A Model of Noisy Software Engineering Data” (Status Report), ACM/IEEE International Conference on Software Engineering, Kyoto, Japan (April 1998) 461-464.

Continued from page 5

- All the types of artifacts in a typical software development lifecycle can be edited using WebDAV, including requirements, design documents, test cases, code, and more. So, while WebDAV provides HTML [7] and XML [2] authoring support, it just as easily supports authoring of existing word processing, spreadsheet, text, graphics, and all other formats.
- WebDAV and HTTP provide a common interface to a wide range of repositories, such as configuration management, file systems, databases, document management, etc. In essence, WebDAV makes the Web look like a large-grain network-accessible file system. But, unlike a conventional file system, a WebDAV-enabled repository provides Internet access, and allows all "files" to be viewed using a standard Web browser.

In November of 1998, the IETF approved the WebDAV Distributed Authoring Protocol [6], clearing the way for broad industry and open source support. Microsoft is a major early supporter of the protocol, providing WebDAV support in Internet Explorer 5 (in its "Web Folders" feature), in Office 2000 (Word and Excel), and on the server side in its Internet Information Services (IIS) server, which comes bundled with every copy of Windows 2000 Server. The popular Apache web server also has WebDAV capability in the freely available mod_dav module. Additional support comes from IBM, Xerox, Novell, DataChannel, CyberTeams, and Digital Creations who have all announced product plans based on WebDAV. WebDAV is clearly a standard that has strong corporate and open source support.

Capabilities

The WebDAV Distributed Authoring Protocol defines a set of extensions to the Hypertext Transfer Protocol (HTTP) for the following capabilities:

- <u>Overwrite prevention:</u> Keeping more than one person from working on a document at the same time. This prevents the "lost update problem" in which modifications are lost as first one developer, then another writes changes without merging the other developer's work.

WebDAV provides facilities for both shared and exclusive locking. This dual lock support provides sufficiently flexible locks to accommodate a wide range of collaborations, with shared locks best supporting collaborators who have a lot of awareness of each other's activities, and exclusive locks providing a more stringent guarantee of conflict avoidance for less aware collaborators, or during periods of high contention for a software development artifact. Locks may have a scope of a single artifact or a hierarchy of artifacts, such as a source code tree. A lock discovery mechanism (a WebDAV property) allows authors to find out if any locks exist on a Web-accessible artifact. Since the Web is designed so that no lock is required to read a Web page, there is no concept of a read lock.

- <u>Properties:</u> Creation, removal, and querying of information about Web-accessible artifacts, such as its author, last modified date, etc. Also included is the ability to make hypertext links between artifacts of any content type.

WebDAV properties are name, value pairs where the name is a Uniform Resource Locator (URL), and the value is a sequence of well-formed Extensible Markup Language (XML) [2] elements. Using URLs as property names provides a globally unique property namespace. Since property names can be URLs, which have a domain name as a component of the URL, property names can be given uniqueness without central registration by using URL property names chosen from within a domain whose name is controlled by the party

Continued on page 7

Continued from page 6

defining the property. So, for example, a company which controls a given domain name, like "widgets.com" can chose a property name from within this domain, like "widgets.com/properties/color".

Using XML to encode the value of properties provides three major benefits. First is extensibility. Since all content within XML is encoded between start and end tags, it is easy to add additional elements to a property by inserting new tagged content. Internationalization is the second major benefit. Since XML mandates support for the UTF-8 and UTF-16 encodings of the ISO 10646 character encoding standard, as well as language tagging, properties can express content in the vast majority of human languages. Finally, by using XML, WebDAV properties can support other metadata activities which are also based on XML, such as the Resource Description Framework (RDF) under development at the W3C.

- <u>Name space management:</u> Creation, removal, and automatic consistency maintenance of collections containing sets of software development artifacts. Also, the ability to copy and move Web-accessible artifacts, and to receive a listing of artifacts in a collection (similar to a directory listing in a file system).

Work-in-Progress

Current work-in-progress within WebDAV focuses on these additional capabilities:

- <u>Version management:</u> The ability to store important revisions of a software artifacts for later retrieval. Version management can also support collaboration by allowing two or more authors to work on the same artifact in parallel tracks. Automatic versioning records successive modifications to an artifact made by versioning unaware ("downlevel") clients. Configuration management tracks versioned collections of versioned artifacts, an important capability for reverting to previously released software packages, or for tracking many resources simultaneously.

A new working group, building on initial design work performed by the WebDAV working group, is currently being formed in the IETF to develop a protocol for Web versioning and configuration management. Called the Delta-V working group, it is already well underway towards its goal of a finished protocol specification in mid-2000. A snapshot of work in progress can be found by examining the group's goals document [10] and the current versioning and configuration management protocol specification [3].

- <u>Advanced Collections:</u> Referential resources act like symbolic links in file systems, while ordered collections maintain a client-specified ordering of resources in a collection without private agreements.
- *Access Control*. The ability to limit the access rights of a given authenticated principal on a given artifact, remotely, via the Internet. WebDAV assumes the existence of, but does not specify, strong authentication technology, and today WebDAV servers have server-specific access controls.

A strongly related effort to WebDAV is the IETF's DAV Searching and Locating (DASL) working group which is developing an interoperable means of searching a repository which is compliant with the WebDAV object model, and which organizes its artifacts into URL hierarchies. The main capability of DASL is searching.

- <u>Searching:</u> Client specified, server-executed queries to locate artifacts based upon their property values and text content. This is an important capability for software development, supporting queries like "where is this function used" or "where is this variable defined".

DASL is working to develop extensions to the WebDAV Distributed Authoring Protocol

Continued on page 8

Continued from page 7

specification (and hence to HTTP) for searching WebDAV repositories. DASL has a requirements document [9] and protocol document [8], which are still the subject of intense effort within the DASL group.

Application

Taken together, the WebDAV extensions to HTTP provide the standard needed to make the Web a writable, collaborative medium. What does this mean for Web-based software development? Although the future is notoriously hard to predict, here are some likely outcomes of the adoption of WebDAV. As WebDAV technology is deployed, it will initially have its largest impact on small to medium sized development teams which support WebDAV, allowing their development practices to coalesce around a local intranet, gaining its advantages for remote access of software development information within the organization. Over time, as critical mass grows, WebDAV will also dramatically reduce the accidental costs of collaboration between development teams, and between development organizations. Since WebDAV supports multiple information repositories, it acts as a common bridge across these stores, providing a convenient mechanism for integrating data in a software development environment. Due to this, WebDAV shows significant promise as an infrastructure for development of distributed software engineering environments, a topic explored in a recent Communications of the ACM article [5].

While WebDAV today significantly reduces the burden of cross-team collaborative software development, the protocol will really start to shine once the Web versioning protocol is finished. Then, remote software engineering teams will be able to remotely edit software artifacts while keeping track of important states of these documents, and while tracking configurations of these versioned objects. Since major configuration management vendors such as Rational, Merant (Intersolv), Microsoft, and IBM are working together to define this standard, broad tool support can be expected.

One of the nice aspects of WebDAV technology is that compelling WebDAV clients and servers are available today, free of charge, ready for initial evaluation. The Apache mod_dav module can be downloaded at: www.webdav.org/mod_dav/, and works with Apache servers 1.3.4 and higher. The Apache server software is freely available for download at: www.apache.org. Or use one of the WebDAV test servers listed at www.webdav.org/projects/. For client software, try Internet Explorer 5's Web Folders feature (IE5 can be downloaded from www.microsoft.com/windows/ie/ie5/default.asp) or download the WebDAV Explorer, a Java-based client developed at the University of California, Irvine from www.ics.uci.edu/~webdav/.

For More Information:

Working groups of the Internet Engineering Task Force are completely open, and may be joined by subscribing to their E-mail discussion list. If you wish to participate in the discussions on WebDAV topics, you may join the mailing list by E-mailing a message with the subject "subscribe" to w3c-dist-auth-request@w3.org.

The home page for the WebDAV group is at URL: www.ics.uci.edu/pub/ietf/webdav/, which contains links to current working drafts, E-mail list archives, and background material. Another excellent source of WebDAV information is the WebDAV Resources page at www.webdav.org/, maintained by Greg Stein. The related DAV Searching and Locating (DASL) working group has its web page at URL: www.ics.uci.edu/pub/ietf/dasl/, and a mailing list which may be joined by sending a message with subject"subscribe" to www-webdav-dasl-request@w3.org.

Continued on page 9

Continued from page 8

About the Author

Emmet James Whitehead is currently a PhD. student at the University of California, Irvine. He has been the Chair of the Internet Engineering Task Force World Wide Web Distributed Authoring and Versioning (WebDAV) Working Group, March 1997-present. With assistance from the World Wide Web Consortium, WebDAV created this working group to be the focus of coordination for the development of interoperability work on tools which support distributed authoring and versioning via the WWW.

Mr. Whitehead's current research focuses on hypertext and software architecture.

Author contact Information

E. James Whitehead, Jr.
Dept. of Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425

ejw@ics.uci.edu
www.ics.uci.edu/~ejw/

References

[1] T. Berners-Lee, R. Fielding, L. Masinter. "Uniform Resource Identifiers (URI): Generic Syntax." Draft Standard Request for Comments (RFC) 2396. MIT/LCS, U.C. Irvine, Xerox. August, 1998. ftp://ftp.isi.edu/in-notes/rfc2396.txt

[2] T. Bray, J. Paoli, C. M. Sperberg-McQueen. "Extensible Markup Language (XML) 1.0," W3C Recommendation, REC-xml, February, 1998. www.w3.org/TR/REC-xml

[3] C. Kaler, J. Amsden, G. Clemm. "Versioning Extensions to WebDAV." Internet-Draft, Work-in-Progress, draft-ietf-webdav-versioning-01, January, 1999. ftp://ftp.isi.edu/internet-drafts/draft-ietf-webdav-versioning-01.txt

[4] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee. "Hypertext Transfer Protocol — HTTP/1.1" Internet Proposed Standard Request for Comments (RFC) 2068. U.C. Irvine, DEC, MIT/LCS. January, 1997. www.ics.uci.edu/pub/ietf/http/rfc2068.txt

[5] R. Fielding, E. J. Whitehead, Jr., K. M. Anderson, G. A. Bolcer, P. Oreizy, R. N. Taylor. "Web-Based Development of Complex Information Products." Communications of the ACM, Vol. 41, No. 8, August, 1998, pages 84-92.

[6] Y. Y. Goland, E. J. Whitehead, Jr., A. Faizi, S. R. Carter, D. Jensen. "Extensions for Distributed Authoring on the World Wide Web — WEBDAV." Proposed Standard Request for Comments (RFC) 2518, February, 1999. ftp://ftp.isi.edu/in-notes/rfc2518.txt

[7] D. Raggett, A. Le Hors, I. Jacobs. "HTML 4.0 Specification" W3C Recommendation REC-html40, April 24, 1998 www.w3.org/pub/WWW/TR/REC-html40.html

[8] S. Reddy, D. Jensen, S. Reddy, R. Henderson, J. Davis, A. Babich. "DAV Searching and Locating." Internet-Draft, Work-in-Progress, draft-reddy-dasl-protocol-04, November, 1998. ftp://ftp.isi.edu/internet-drafts/draft-reddy-dasl-protocol-04.txt

[9] J. Davis, S. Reddy, J. Slein. "Requirements for DAV Searching and Locating." Internet-Draft, Work-in-Progress, draft-ietf-dasl-requirements-00, February, 1999. ftp://ftp.isi.edu/internet-drafts/draft-ietf-dasl-requirements-00.txt

[10] J. Stracke, J. Amsden. "Goals for Web Versioning." Internet-Draft, Work-in-Progress, draft-ietf-webdav-version-goals-00, February, 1999. ftp://ftp.isi.edu/internet-drafts/draft-ietf-webdav-version-goals-00.txt

[11] J. Slein, F. Vitali, E. J. Whitehead, Jr., D. Durand. "Requirements for Distributed Authoring and Versioning on the World Wide Web." Xerox, University of Bologna, U.C. Irvine, Boston University. Internet Informational Request for Comments (RFC) 2291, February, 1998. www.ics.uci.edu/pub/ietf/webdav/requirements/rfc2291.txt

Orbit: A Next Generation Collaboration Environment

James Milligan & Carla Burns, Air Force Research Laboratory Information Technology Division

Introduction

The Information Age of this century has allowed communication technology to boldly go well beyond those of the past. Cellular phones, beepers, personal workstations, computer networks, and the Internet have all contributed to the Information Age. Some of today's technologies are aimed at providing collaboration support for the workday activities of distributed groups. These activities may include communication, cooperation, problem solving and/or negotiation and may be directed towards almost any application including telemedicine, Command and Control, and software development/evolution. While existing collaboration environments have come a long way in providing capabilities to support group work, issues such as usability, flexibility and interoperability remain. The Orbit work aims to overcome such obstacles through the development of a next generation collaboration environment. Three key elements contribute to Orbit's approach for a next generation collaboration environment:

1) a theoretical framework based on groupwork in complex domains;
2) a multi-functional, open architectural framework for support of collaboration activities and the theoretical framework; and
3) powerful data visualization facilities which improve user understanding.

The Theoretical Framework of Orbit: Locales.

Orbit's theoretical framework is based upon the work of sociologist Anselm Strauss and his notion of social worlds [1, 2]. Strauss' social world model addresses the understanding, analysis and reasoning about group work in complex domains. The Orbit work takes many of the social concepts in Strauss' model and applies them to support collaboration activities through networks of computers using locales. Locales are virtual places for group work situated in the computer network. Strauss' work was selected as the theoretical basis for Orbit due its support of the following group work concepts: displaced action as the central focus of work, admitted flexibility and contingency of work and equal weight of formal and informal aspects of work.

The locales framework integrates five principles necessary for distributed group collaboration. These principles are locale foundations, mutuality, individual views, interaction trajectories, and civic structures. A locale can be thought of as a focal point around which to define, structure, and relate the relevant people, objects, tools, and resources germane to a particular collaboration activity. The locales foundations principle captures the basic domain structuring and furnishing of work. Locale foundations are therefore about a) providing adequate media and mechanisms in available domains to support the sharing of objects, tools, and resources, b) supporting a group's notion of membership and related processes, and c) facilitating appropriate privacy and access mechanisms.

The civic structure principle deals with facilitating interaction with the wider community beyond an individual's immediate workgroups and locales. It includes the lifecycle processes that support the emergence and dissolution of locales and the structuring of the world of locales in the broader sense. This is where external influences beyond the locales of direct interest can be considered (e.g. organizational, professional, financial, and political).

The trajectory principle is concerned with all of the temporal aspects of the group's locale, its associated individuals and entities. It also deals with the phasing, articulation and management of interactions. The mutuality principle concerns the

Continued on page 11

Continued from page 10

degree to which awareness and presence must be supported in the collaborative work. The mutuality principle is important for both synchronous and asynchronous interactions, although the medium of expression might be very different.

Finally, the individual view principle looks at the different individual views of the same locale and the individual views over multiple locales. While there may be a group definition of the locale, the individuals in the group may all have a different view of, or interest in the locale based on their current level of involvement. Moreover, few individuals have the luxury of being able to focus on a single task exclusively. They usually belong to multiple social worlds and work on many different tasks concurrently, with varying (and shifting) degrees of intensity.

The Orbit Environment: Multi-functional and Open Architecture

The Orbit environment provides collaboration tools that support the locales theoretical framework which is guided by the five principles discussed above. It allows spontaneous generation and evolution of networks of locales. Each locale is dedicated to a particular purpose and furnished with the artifacts and tools required for distributed workgroups to effectively accomplish the task at hand. When groups are working they need the following [3]:

- The family of artifacts that make up the formal layer of their work activities. Examples include program files, medical records, yellow stickies, etc.
- The tools that are used to manipulate these artifacts. Examples include compilers, editors, pens, etc.
- Resources for effective communication that grant members of the social world the ability to communicate appropriately to the task at hand.
- Automation of mundane tasks, such as change notification.
- The ability to navigate. That is, to seamlessly switch among multiple on-going tasks, interrelate them as appropriate, and find tasks and people as needed.

The environment emphasizes flexible coordination and communication, and can be easily integrated with workgroup repositories, artifacts and tools as well as other workgroup management systems. Key features of the environment include a ubiquitous collaborative desktop, wide-area scalable collaboration infrastructure, synchronous and asynchronous group support, user-controlled projection of presence and awareness and pervasive audio/video capabilities. Figure 1 provides an illustration of the Orbit environment. The Orbit user interface shown in the figure,

Continued on page 12

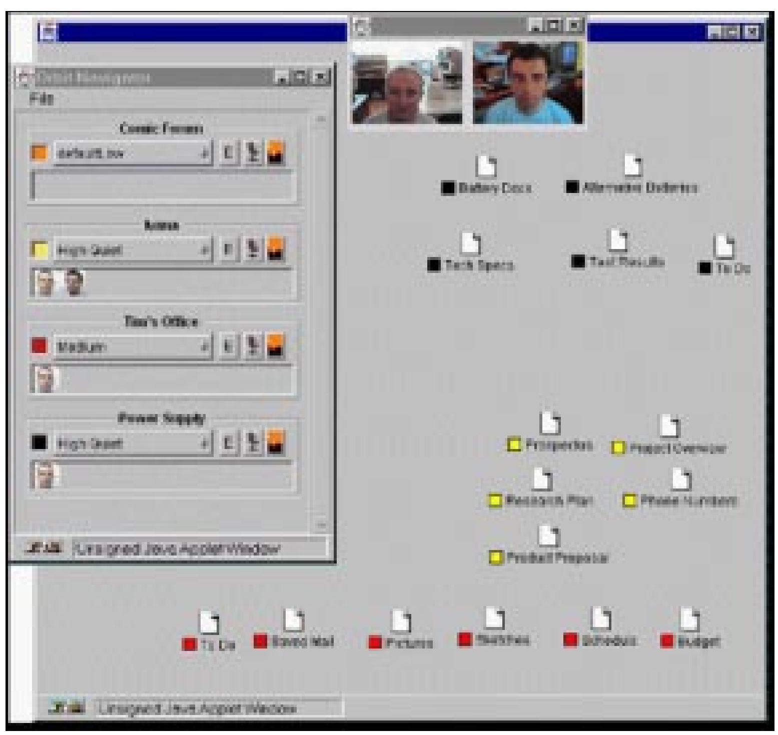

Figure 1. The Orbit Collaboration Environment

Continued from page 11

consists of a locale navigator and a locale workspace. The navigator permits configuration of locales, objects, documents and shared audio/video conferencing. The workspace provides capabilities for viewing and manipulating shared repository artifacts.

Orbit is based on a three-layer conceptual model that implements locales and individual views based on the external objects in the lowest layer. The locales layer groups objects and tools together and provides support for presence, awareness and trajectory information. The individual views layer provides the user with vision into multiple locales simultaneously and with varying degrees of intensity. The system provides a ubiquitous collaborative desktop through which users can perform all shared and individual tasks. Figure 2 presents an illustration of Orbit's conceptual architecture

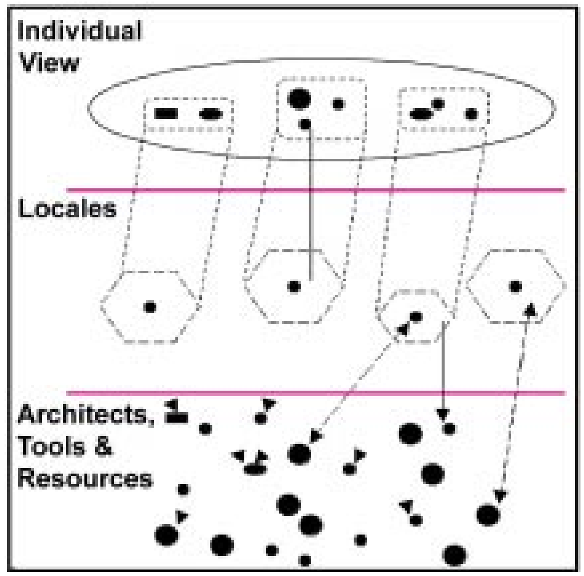

Figure 2. Orbit Conceptual Architecture

The Data Visualization Capabilities of Orbit: Virtue

In the physical world, individuals react directly with their daily environments. The Orbit environment provides powerful data visualization capabilities that immerse the individual into his/her domain. This component is called Virtue. To date, Virtue has been applied to the system engineering arena by providing a direct manipulation interface for exploring software structure, evolution, and behavior dynamics. Key capabilities of the Virtue component include shared 3D views of software structure and history, virtual tools for exploring data visualizations, multimedia annotations for spatio-temporal marking, tactile feedback for grasping and manipulation and voice command for unencumbered interaction. The actions of each Virtue user are reflected in all other coupled virtual spaces. Future work will explore the application of Virtue to other application domains such as telemedicine and Command and Control.

Acknowledgments

The Orbit and Virtue work is currently being sponsored by the DARPA Evolutionary Design of Complex Systems (EDCS) Program and the Air Force Research Laboratory (AFRL) Information Technology Division. Additional sources of sponsorship for the work have been provided by the DARPA Intelligent Collaboration & Visualization and Software Environments Programs, the US Army Construction Engineering Laboratory, the National Science Foundation, Sun Microsystems, Hewlett-Packard Laboratories, Digital Equipment Corporation, Intel and Fujitsu Open Systems Solutions.

Continued on page 13

Continued from page 12

Additional Information

Further information on Orbit can be found at: www.dstc.edu.au/worlds

Further information on Virtue can be found at:
www-pablo.cs.uiuc.edu/Project/VR/VirtueOverview.html

About the Authors

James Milligan has been a computer scientist with Air Force Research Laboratory (AFRL) for 14 years where he has directed research and development activities in software engineering. He is currently the AFRL Program Manager for the Evolutionary Design of Complex Software (EDCS) Program. He received his B.S. in Computer Science from the State University of New York (SUNY) School of Technology, and is currently finalizing his M.S. in Software Development and Management through Rochester Institute of Technology.

Carla Burns has been a computer scientist with Air Force Research Laboratory (AFRL) for 13 years where she has directed research and development activities in software engineering in the areas of requirements engineering, rapid prototyping, and scenario generation. She received her B.S. in Mathematics from Clarkson University and a M.S. in Computer Science from the State University of New York (SUNY) Institute of Technology.

Author Contact Information

James Milligan

Air Force Research Laboratory
AFRL/IFTD
525 Brooks Road
Rome, NY 13441-4505

milliganj@rl.af.mil

Carla Burns

Air Force Research Laboratory
AFRL/IFTD
525 Brooks Road
Rome, NY 13441-4505

burnsc@rl.af.mil

References

[1] Fitzpatrick, G., W.J. Tolone, et al. (1995). Work, Locales and Distributed Social Worlds.

[2] Strauss, A. (1993). Continual Permutations of Action. New York, Adeline De Gruyter.

[3] University of Illinois (1998). “Draft Final Technical Report for Atlantis”, Air Force
Contract No. F30602-94-0161.

[4] Orbit Brochure; Kaplan, Simon; Reed, Daniel (1998). Evolutionary Design of Complex Systems Demonstration Days

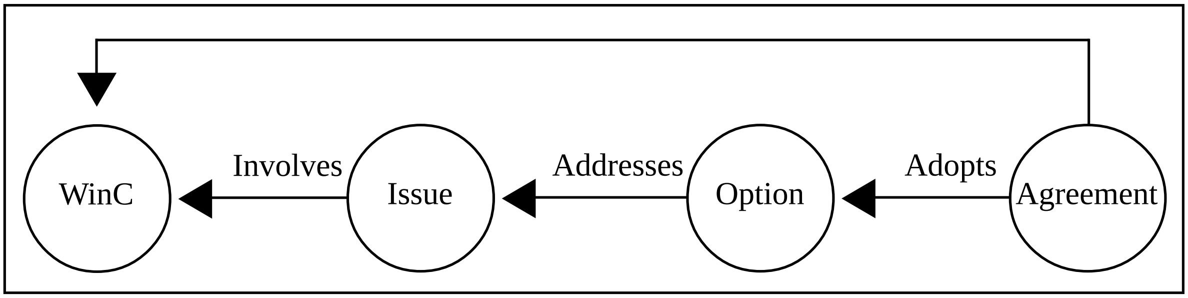

Figure 1. A Simple Scenario

1. An owner of the project is identified. He identifies the other people who will participate in this negotiation.
2. The owner starts WinWin, creates the new project, and enters the names of all users. These people are called stakeholders.
3. One stakeholder is designated to define or tailor an existing set of terms for the proposed system. He enters them in WinWin.
4. One stakeholder is designated to define or tailor an existing taxonomy for the proposed system. He enters the taxonomy in WinWin.
5. Stakeholders review and iterate the terms and taxonomy.
6. One now begins the negotiation process which continuously loops through steps 6 and 7:

   6a. stakeholders create Win Conditions expressing their preferences, and/or

   6b. stakeholders create Issues that they believe exist, and/or

   6c. stakeholders create proposed Agreements.
7. Stakeholders review newly entered artifacts with existing artifacts.

   7a. a new conflict is observed so a new Issue is created

   7b. stakeholders develop Options to address Issues

   7c. stakeholders create new Agreements and vote on Agreements in-Progress.

Steps 6 and 7 continue until all Win Conditions are covered, all Issues are resolved and all Agreements are passed. Win Conditions and Issues that no longer are relevant, Options that are unused, and Agreements that have failed are marked as INACTIVE. Inactive artifacts are not shown, by default, but there is a way to restore them.

A More Complicated Example of Negotiation

(Illustrated in Figure 2 on page 15.)

Suppose we have two win conditions involved in an issue. The issue has one option which is adopted by Agreement1. Agreement1 is voted on, passes and in turn covers Win Condition 1 and 2. Now suppose a new win condition is entered which causes Agreement1 to become invalid. What are the actions that should result?

1. The owner of Agreement1 changes its status to INACTIVE. This causes Option1 to be unadopted, Issue1 to be unresolved, and Win Conditions 1 and 2 to be uncovered.
2. A new issue is drafted, Issue2, which involves Win Conditions 1, 2, and 3. Issue2 replaces Issue1.
3. Options to resolve Issue2 are generated and Option2 is chosen to create an agreement, Agreement2. Agreement2 replaces Agreement1.
4. Agreement2 initiates a vote which eventually passes. This causes Option2 to be adopted, Issue2 resolved, and Win Conditions 1, 2, and 3 to be covered.

Continued on page 15

Continued from page 15

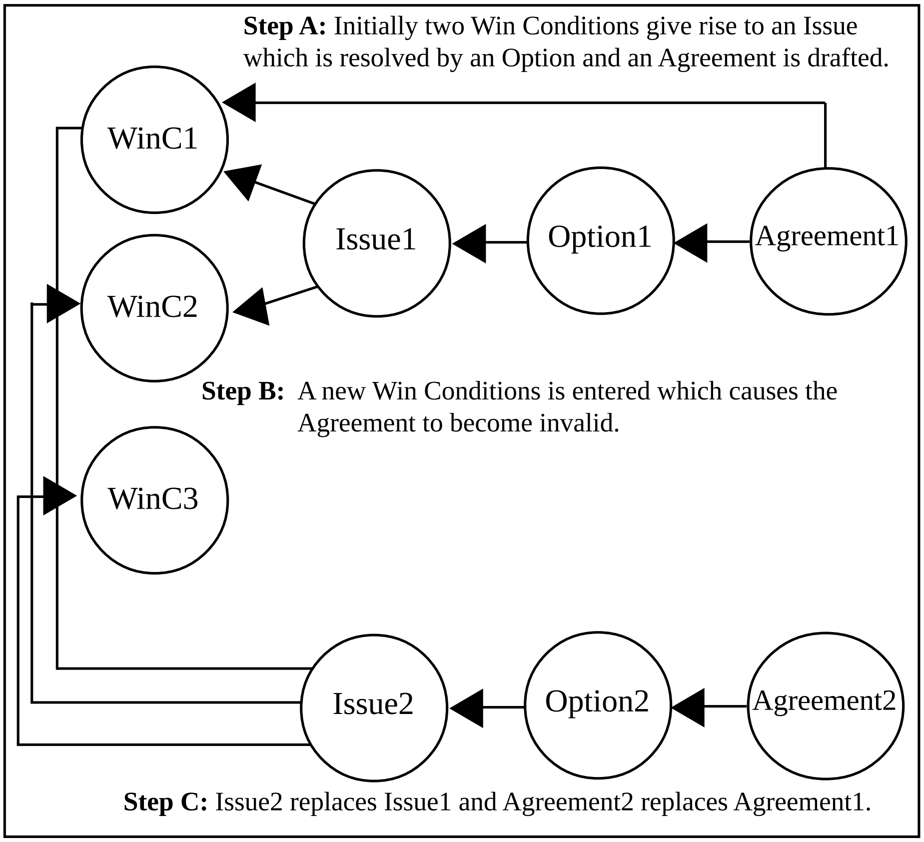

Figure 2. A More Complicated Negotiation

Rationale Graph

One of the essential elements in any negotiation is a record of the arguments that were used in favor or against a particular issue or option. WinWin assists in the capture and retention of all such arguments through its process model described earlier and a rationale graph. In technical terms, a rationale graph is the transitive closure of the set of nodes that are reachable from an Agreement. In effect this includes all proposed Options, whether adopted or not, all Issues which are eventually addressed and all Win Conditions. This graph is displayed by the program as an indented list. By tracing through the web of interconnections one may completely resurrect the arguments that were used which led to the adoption of the Agreement.

In the Figure 3 is a picture of artifact customer-AGRE-1. This is an Agreement artifact. On the right hand side you see the Artifact Set window. This agreement points to an option, customer-OPTN-1, which in turn points to an issue, customer-ISSU-1. The agreement also points to a win condition, user-WINC-3. At some later point in the process, stakeholders will vote on this Agreement. Once a vote is started, all pointers to artifacts are frozen, as the artifacts must maintain the identical form throughout the voting process. Once a vote is complete, the Agreement either passes or fails.

Another form of rationale support in WinWin is the Rationale field. This field is placed next to the body description of an artifact, and can be seen in Figure 4. The stakeholder may explicitly provide his rationale for a particular artifact by entering a statement in that field.

Win Win Attachments

WinWin recognizes that there may be auxiliary tools that stakeholders desire to use during the course of a negotiation. For example one might use a spreadsheet to analyze the financial impacts of a given Option. Or one might use a program such as COCOMO to estimate the effort and schedule required for a particular decision. WinWin provides a capability to attach such programs and their outputs to any artifact in the system. This is called the Attachment field. By making the Attachment field be a part of every artifact, stakeholders may associate these program elaborations at a desirable level of granularity. The Attachment field allows for an arbitrary number of attachments. Each attachment includes the name of the program and its associated data set.

Continued on page 16

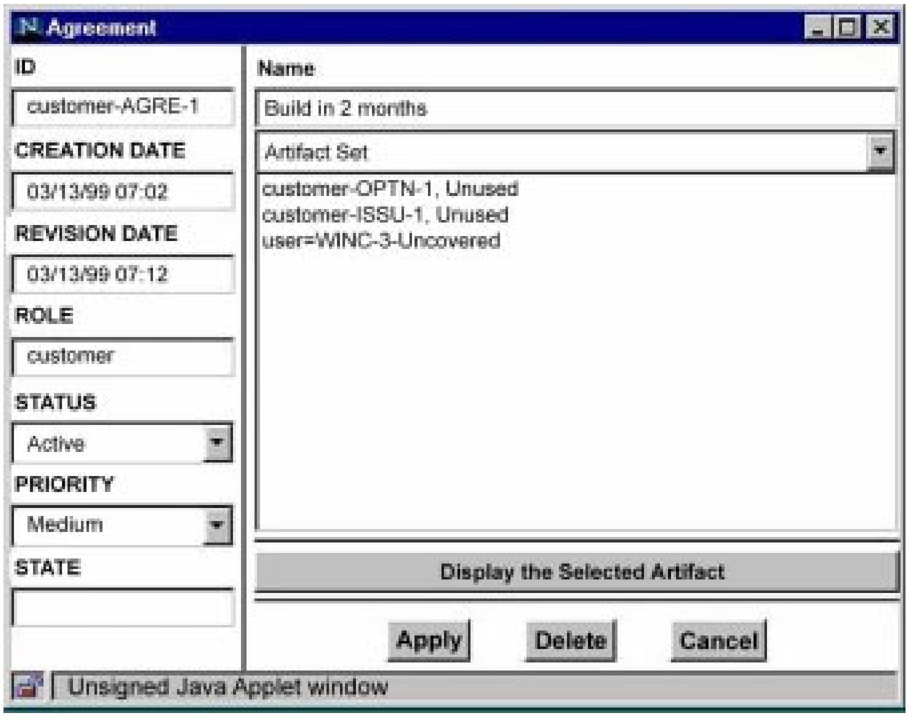

Figure 3. Agreement Artifact

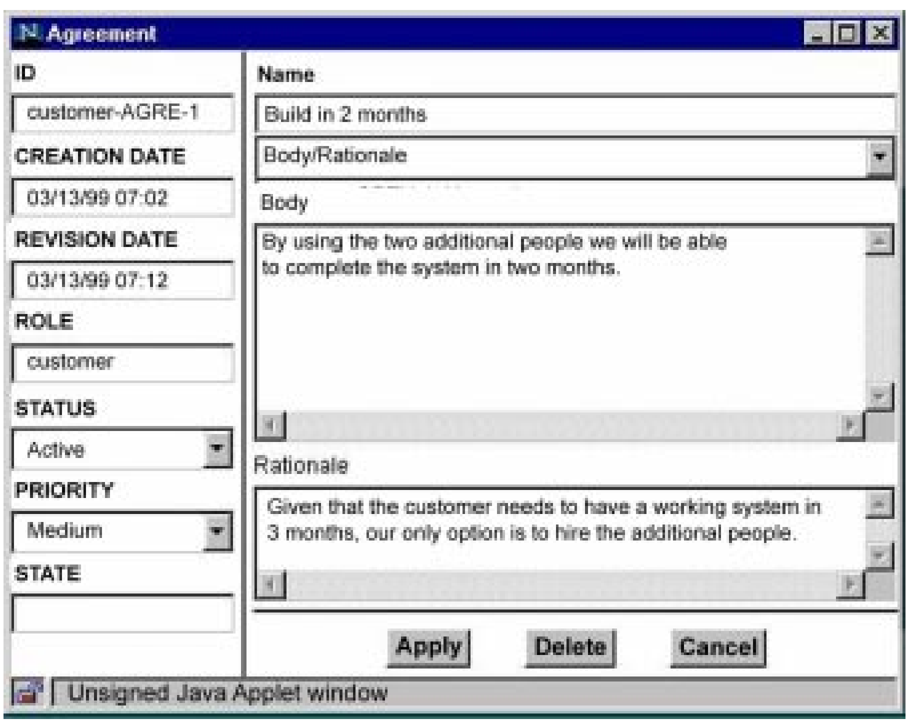

Figure 4. Rationale Field

Figure 5 shows an example of the Attachments field of an artifact. On the right hand side of the window, towards the bottom are two input lines, labeled Tool and File. Here is placed the name of the program to be attached plus any associated data file. When the apply button is clicked, the program name and data file are placed in the window immediately above, with all other attachments.

Mode of Operation

As a tool for requirements capture and negotiation, WinWin assumes that stakeholders will be potentially working at different locations and at different times. Thus WinWin supports a distributed, asynchronous mode of operation. A stakeholder may sign onto the system at any time. There may or may not be other stakeholders using the system. The stakeholder may examine the Messages, a record of all changes made to the WinWin database by the stakeholders. These messages are ordered by date, and each stakeholder has the option to maintain or discard any or all of the Messages.

Figure 6 on page 18, shows some sample output of the Messages window. Each artifact is named by its unique identifier, e.g., user-TERM-1 is the artifact that belongs to the stakeholder named user and it is an artifact of type TERM. Each line in the Messages window refers to a

Continued on page 17

Continued from page 16

unique action performed by the stakeholder. For example, the first line in the figure indicates that a new TERM artifact has been created on 02/27/98 at time 19:22. Other messages indicate the date and time that the artifact was modified, including the name of the field that was modified. At the bottom of this window there are three buttons. The Delete button will remove the highlighted Message. Cancel causes the Message window to disappear. OK causes the artifact in the highlighted line to appear.

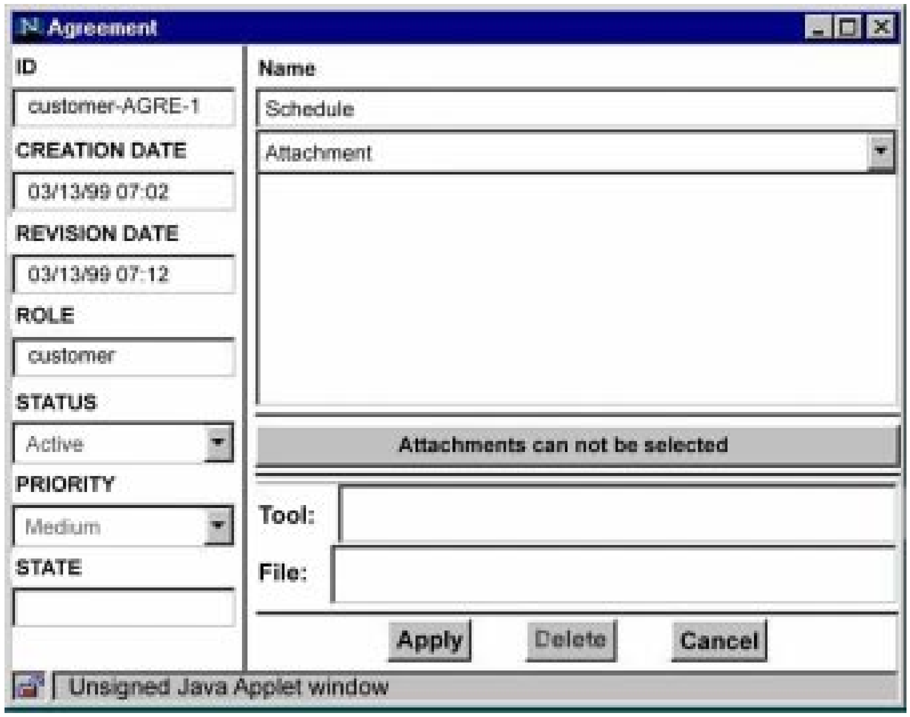

Figure 5. Attachments Field

Win Win Versions

Win Win was developed in C, X-Windows and Motif and runs on Solaris, HP-UX and Linux operating systems. It is available from: http://sunset.usc.edu/WinWin/winwin.html#download

There is a version of Win Win that has been developed using Java. This version can be run from the CSE web site by invoking the URL: http://sunset.usc.edu/javawinwin/winwin.html

You may download the Java class files and install Java Win Win at your own site. To do this invoke the URL: http://sunset.usc.edu/jwins.html

Win Win API

We have developed a library of functions which can be used to create programs that interact directly with WinWin. This is referred to as the WinWin Application Programmers Interface or WinWin API. In addition to the library we distribute a test program. This program should be run after WinWin is installed to make sure the API is functioning properly. In addition we have provided source for the test program so people interested in using the API can imitate this program.

Analysis of the Win Win Process Model

(Illustrated in Figure 7 on page 19)

Figure 7 shows a state transition diagram that describes the various states of the WinWin database as negotiation proceeds. Nodes describe the possible states of the database while transitions are actions taken by the WinWin system or by the stakeholders.

Acknowledgments

Thanks go to everyone at the Center for Software Engineering who participated in the design and development of WinWin. That includes Prasanta Bose, Ming June Lee, Ahmed Abdullah, Cristina Gacek, Anne Curran, and Barry Boehm. Also to our industrial affiliates who contributed time and effort in testing out earlier versions. Their contributions are gratefully acknowledged.

Continued on page 18

Continued from page 17

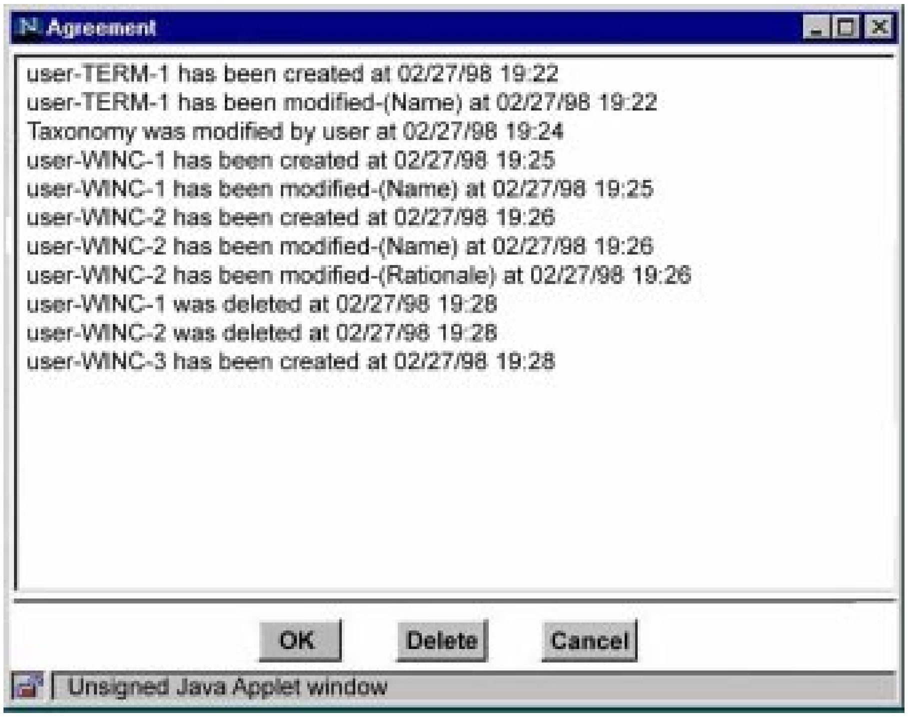

Figure 6. Message Window

Author Contact Information

Ellis Horowitz
Center for Software Engineering
& Computer Science - U.S.C.
Los Angeles, CA 90089
horowitz@usc.edu

Joo H. Lee
Center for Software Engineering
& Computer Science - U.S.C.
Los Angeles, CA 90089
joolee@sunset.usc.edu

June Sup Lee
Center for Software Engineering
& Computer Science - U.S.C.
Los Angeles, CA 90089
junelee@sunset.suc.edu

References

[1] B. Boehm and H. In, “Aids for Identifying Conflicts Among Quality Requirements,” One of best papers at *International Conference on Requirements Engineering* (ICRE96), Colorado, April 1996, and *IEEE Software*, March 1996

[2] P. Bose, “Conceptual Design Model based Requirements Analysis in the WinWin Framework for Concurrent Requirements Engineering”, *IWSSD*, 1995

[3] P. Bose, “A Model for Decision Maintenance in the WinWin Collaboration Framework”, *Knowledge Based Software Engineering* (KBSE '95), 1995

[4] B. Boehm, P. Bose, E. Horowitz, and M. J. Lee, “Software Requirements Negotiation and Renegotiation Aids: A Theory-W Based Spiral Approach,” *Proceedings of the 17th International Conference on Software Engineering* (ICSE-17), Seattle, April 1995.

[5] M. J. Lee, “Formally Modeling the WinWin Requirements Negotiation System,” Doctoral Consortium, *Second IEEE International Symposium on Requirements Engineering* (RE’95), York, March 1995.

[6] M. J. Lee, and Barry Boehm, Knowledge Summary: *USC Center for Software Engineering Focused Workshop on Next Generation Software Processes and Their Support*, USC Center for Software Engineering, 1994.

[7] B. Boehm and P. Bose, “A Collaborative Spiral Software Process Model Based on Theory W”, *Third International Conference on the Software Process*, 1994.

[8] B. Boehm, P. Bose, E. Horowitz, and M. J. Lee, “Experimental Results from a Prototype Next-Generation Process Support System,” *TRW Systems Integration Group Technology Review*, Summer 1994, Volume 2, Number 1.

[9] B. Boehm, P. Bose, E. Horowitz and M. J. Lee, “Software Requirements As Negotiated Win Conditions,” Proceedings of the First International Conference on Requirements Engineering (ICRE94), IEEE Computer Society Press, Colorado Springs Colorado, April 1994.

[10] CSE Web Site and all software can be found at: http://sunset.usc.edu/.

Continued on page 19

Continued from page 18

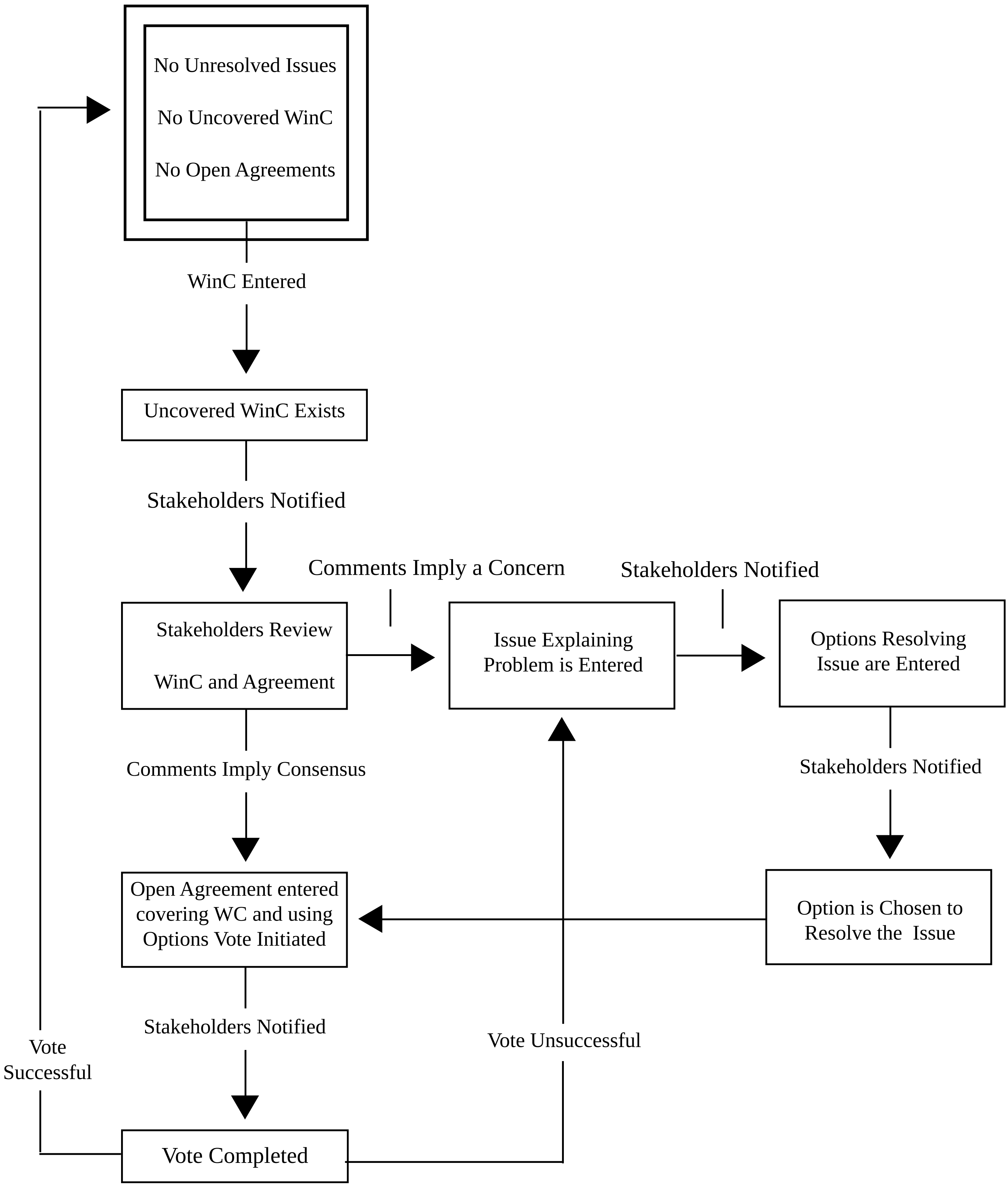

Figure 7. WinWin Transition States

DoD DACS Products & Services Order Form

Name: Position/Title:

Organization: Acronym:

Address:

City: State: Zip Code:

Country: E-mail:

Telephone: Fax:

Product Description	*Format*	*Quantity*	*Price*	*Total*
The DACS Information Package	***Note: All Disks are available in PC or Mac**			
❑ Including: 2 recent Software Tech News newsletters, and several DACS Products & Services Brochures	Documents		FREE	FREE
Empirical Data				
❑ Architecture Research Facility (ARF) Error Dataset	Disk		$ 50	
❑ NASA / Software Engineering Laboratory (SEL) Dataset	CD-ROM		$ 50	
❑ NASA / AMES Error/Fault Dataset	Disk		$ 50	
❑ Software Reliability Dataset	Disk		$ 50	
❑ DACS Productivity Dataset	Disk		$ 50	
Technical Reports				
❑ A Business Case for Software Process Improvement	Document	FREE with Spreadsheet	$ 25	
❑ ROI from Software Process Improvement Spreadsheet	Disk		$ 40	
❑ A History of Software Measurement at Rome Laboratory	Document		$ 25	
❑ An Analysis of Two Formal Methods: VDM and Z	Document		$ 25	
❑ An Overview of Object-Oriented Design	Document		$ 25	
❑ Artificial Neural Networks Technology	Document		$ 25	
❑ A Review of Formal Methods	Document		$ 25	
❑ A Review of Non-Ada to Ada Conversion	Document		$ 25	
NEW! ❑ A State of the Art Report: Using Defect Tracking & Analysis	Document		$ 50	
❑ A State of the Art Report: Software Design Methods	Document		$ 25	
❑ A State of the Art Review: Distributable Database Technology	Document		$ 25	
❑ Electronic Publishing on the World Wide Web: An Engineering Approach	Document	SALE Item!	**$ 5**	
NEW! ❑ Object Oriented Database Management Systems (Revisited)	Document		$ 50	
❑ Software Analysis and Testing Technologies	Document		$ 25	
❑ Software Design Methods	Document		$ 25	
❑ Software Prototyping and Requirements Engineering	Document		$ 25	
❑ Software Interoperability	Document		$ 25	
❑ Software Reusability	Document		$ 25	
NEW! ❑ A State of the Art Report: Understanding & Improving Technology Transfer in Software Engineering	Document		$ 50	
Bibliographic Products				
❑ Rome Laboratory Research in Software Measurement	Document		$ 25	
❑ DACS Custom Bibliographic Search	Disk		$ 40	
❑ DACS Software Engineering Bibliographic Database (SEBD)	CD-ROM		$ 50	

Method of Payment:

❑ Check ❑ Mastercard ❑ Visa

Number of Items Ordered ____ *Total Cost* ____

Credit Card # ______________________ Expiration Date ______________________

Name on Credit Card ______________________ Signature ______________________

Mail this form to: DACS Customer Liaison
DoD Data & Analysis Center for Software
P.O. Box 1400, Rome, NY 13442-1400

Telephone: (315) 334-4905
Fax: (315) 334-4964
E-mail: **cust-liasn@dacs.dtic.mil**

This form is also on-line at: www.dacs.dtic.mil/forms/orderform.shtml

Software Tech News on the World Wide Web

This newsletter in its entirety available on the web at:
www.dacs.dtic.mil/awareness/newsletters/listing.shtml

Other Collaborative Techniques Web Resources

DoD DACS Collaborative Techniques Topic Area - www.dacs.dtic.mil

Association for Computing Machinery (ACM) Special Interest Group on Supporting Group Work (SIG GROUP) - www.acm.org/siggroup/

Collaborative Strategies - www.collaborate.com

Columbia University Prototype, WebDAV - www.cs.columbia.edu/~eaddy/webdav.html

Computer Supported Collaborative Work (CSCW) Systems -
www.cs.waikato.ac.nz/cs/Research/cscw/ and www.csc.liv.ac.uk/~team-it/index.html

ICSE 98 Workshop on Software Engineering over the Internet: A List of all accepted & presented submissions - http://sern.cpsc.ucalgary.ca/~maurer/ICSE98WS/ICSEWSSubmissions.html

Orbit Collaborative Environment - www.dstc.edu.au/worlds

University of California, Irvine, WebDAV - www.ics.uci.edu/~webdav/
University of California, Irvine, IETF WEBDAV Working Group -
www.ics.uci.edu/pub/ietf/webdav/

Web-Based Distributed Authoring and Versioning (WebDAV) Resourses - www.webdav.org

WebME - the WebME system talked about in this newsletter is available in prototype form at:
ftp://ftp.cs.umd.edu/pub/sel/roseanne/webme

WebSoft - using the Web as the infrastructure and external integration mechanism for a global software engineering environment - www.ics.uci.edu/pub/websoft/

WinWin Spiral Model - http://sunset.usc.edu/WinWin/winwin.html

---

DoD Data & Analysis Center for Software
P.O. Box 1400
Rome, NY 13442-1400

First-Class Mail
U.S. Postage

Return Service Requested